

Please type a plus sign (+) inside this box → ☐

PTO/SB/05 (4/98)
Approved for use through 09/30/2000. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UTILITY PATENT APPLICATION TRANSMITTAL <small>(Only for new nonprovisional applications under 37 C.F.R. § 1.53(b))</small>	Attorney Docket No.	P3815
	First Inventor or Application Identifier	Mario Nemirovsky et al.
	Title	Methods and Apparatus for Managing a Buffer of Events in the Background
	Express Mail Label No.	EL573442895US

APPLICATION ELEMENTS <small>See MPEP chapter 600 concerning utility patent application contents.</small>	ADDRESS TO: Assistant Commissioner for Patents Box Patent Application Washington, DC 20231
--	--

- | | |
|--|---|
| 1. <input checked="" type="checkbox"/> * Fee Transmittal Form (e.g., PTO/SB/17)
<small>(Submit an original and a duplicate for fee processing)</small>
2. <input checked="" type="checkbox"/> Specification [Total Pages 28]
<small>(preferred arrangement set forth below)</small>
- Descriptive title of the Invention
- Cross References to Related Applications
- Statement Regarding Fed sponsored R & D
- Reference to Microfiche Appendix
- Background of the Invention
- Brief Summary of the Invention
- Brief Description of the Drawings (if filed)
- Detailed Description
- Claim(s)
- Abstract of the Disclosure
3. <input checked="" type="checkbox"/> Drawing(s) (35 U.S.C. 113) [Total Sheets 5]
4. Oath or Declaration [Total Pages 2]
a. <input checked="" type="checkbox"/> Newly executed (original or copy)
b. <input type="checkbox"/> Copy from a prior application (37 C.F.R. § 1.63(d))
<small>(for continuation/divisional with Box 16 completed)</small>
i. <input type="checkbox"/> DELETION OF INVENTOR(S)
Signed statement attached deleting inventor(s) named in the prior application, see 37 C.F.R. §§ 1.63(d)(2) and 1.33(b). | 5. <input type="checkbox"/> Microfiche Computer Program (Appendix)
6. Nucleotide and/or Amino Acid Sequence Submission
<small>(if applicable, all necessary)</small>
a. <input type="checkbox"/> Computer Readable Copy
b. <input type="checkbox"/> Paper Copy (identical to computer copy)
c. <input type="checkbox"/> Statement verifying identity of above copies |
|--|---|

*** NOTE FOR ITEMS 1 & 13: IN ORDER TO BE ENTITLED TO PAY SMALL ENTITY FEES, A SMALL ENTITY STATEMENT IS REQUIRED (37 C.F.R. § 1.27), EXCEPT IF ONE FILED IN A PRIOR APPLICATION IS RELIED UPON (37 C.F.R. § 1.28).**


ACCOMPANYING APPLICATION PARTS	
7. <input type="checkbox"/> Assignment Papers (cover sheet & document(s)) 8. <input type="checkbox"/> 37 C.F.R. § 3.73(b) Statement <input checked="" type="checkbox"/> Power of Attorney <small>(when there is an assignee)</small> 9. <input type="checkbox"/> English Translation Document (if applicable) 10. <input type="checkbox"/> Information Disclosure Statement (IDS)/PTO-1449 <input type="checkbox"/> Copies of IDS Citations 11. <input type="checkbox"/> Preliminary Amendment 12. <input checked="" type="checkbox"/> Return Receipt Postcard (MPEP 503) <small>(Should be specifically itemized)</small> 13. <input checked="" type="checkbox"/> * Small Entity Statement(s) <input type="checkbox"/> Statement filed in prior application, Status still proper and desired <small>(PTO/SB/09-12)</small> 14. <input type="checkbox"/> Certified Copy of Priority Document(s) <small>(if foreign priority is claimed)</small> 15. <input checked="" type="checkbox"/> Other: <u>Check for fees</u>	


16. If a CONTINUING APPLICATION, check appropriate box, and supply the requisite information below and in a preliminary amendment:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No: _____

Prior application information: Examiner _____ Group / Art Unit: _____

For CONTINUATION or DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 4b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

17. CORRESPONDENCE ADDRESS			
<input checked="" type="checkbox"/> Customer Number or Bar Code Lab		<input type="checkbox"/> Correspondence address below	
			
Name	24739		
PATENT TRADEMARK OFFICE			
Address			
City	State	Zip Code	
Country	Telephone	Fax	

Name (Print/Type)	Donald R. Boys	Registration No. (Attorney/Agent)	35074
Signature		Date	06/30/2000

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

Certificate of Express Mailing

JCS15 U.S. PTO
09/608750
06/30/00

"Express Mail" Mailing Label Number: EL573442895US

Date of Deposit: 06/30/2000

Ref: Case Docket No.: P3815

First Named Inventor: Mario Nemirovsky et al.

Serial Number: NA

Filing Date: 06/30/2000

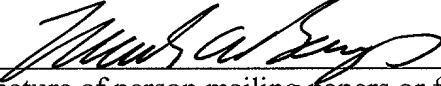
Title of Case: Methods and Apparatus for Managing a Buffer of Events in the Background

I hereby certify that the attached papers are being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. 1.10 on the date indicated above and addressed to the Commissioner of Patents and Trademarks, Washington D.C. 20231

1. Utility patent application transmittal.
2. 28 sheets of specification.
3. 5 sheets of drawings.
4. Fee transmittal.
5. Duplicate fee transmittal.
6. Declaration and Power of Attorney.
7. Verified Statement Claiming Small Entity.
8. Check for fees in the amount of \$483.00.
9. Certificate of express mailing.
10. Postcard listing contents.

Mark A. Boys

(Typed or printed name of person mailing paper or fee)


(Signature of person mailing papers or fee)

PTO/SB/10 (10-96)
 Approved for use through 10/31/99 OMB 0581-0031
 Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
 Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

**VERIFIED STATEMENT CLAIMING SMALL ENTITY STATUS
 (37 CFR 1.9(f) & 1.27(c))--SMALL BUSINESS CONCERN**

Docket Number (Optional)
P3815

Applicant or Patentee: Mario Nemirovsky, et al.

Application or Patent No.: NA

Filed or Issued: NA

Title: Methods and Apparatus for Managing a Buffer of Events in the Background

I hereby declare that I am

- ☐ the owner of the small business concern identified below
☒ an official of the small business concern empowered to act on behalf of the concern identified below.

NAME OF SMALL BUSINESS CONCERN XStream Logic, Inc.

ADDRESS OF SMALL BUSINESS CONCERN 750 University Ave., Suite 270, Los Gatos, CA 95032-7698

I hereby declare that the above identified small business concern qualifies as a small business concern as defined in 13 CFR 121.12, and reproduced in 37 CFR 1.9(d), for purposes of paying reduced fees to the United States Patent and Trademark Office, in that the number of employees of the concern, including those of its affiliates, does not exceed 500 persons. For purposes of this statement, (1) the number of employees of the business concern is the average over the previous fiscal year of the concern of the persons employed on a full-time, part-time, or temporary basis during each of the pay periods of the fiscal year, and (2) concerns are affiliates of each other when either, directly or indirectly, one concern controls or has the power to control the other, or a third party or parties controls or has the power to control both.

I hereby declare that rights under contract or law have been conveyed to and remain with the small business concern identified above with regard to the invention described in:

- ☒ the specification filed herewith with title as listed above
☐ the application identified above
☐ the patent identified above

If the rights held by the above identified small business concern are not exclusive, each individual concern, or organization having rights in the invention must file separate verified statements averring to their status as small entities, and no rights to the invention are held by any person, other than the inventor, who would not qualify as an independent inventor under 37 CFR 1.9(c) if that person made the invention, or by any concern which would not qualify as a small business concern under 37 CFR 1.9(d), or a nonprofit organization under 37 CFR 1.9(e)

- ☒ Each person, concern, or organization having any rights in the invention is listed below.
☐ no such person, concern, or organization exists.
☐ each such person, concern, or organization is listed below.

Separate verified statements are required from each named person, concern or organization having rights to the invention averring to their status as small entities. (37 CFR 1.27)

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate (37 CFR 1.28(b))

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

NAME OF PERSON SIGNING Mario Nemirovsky

TITLE OF PERSON IF OTHER THAN OWNER Chief Technical Officer

ADDRESS OF PERSON SIGNING 750 University Ave., Suite 270, Los Gatos, CA 95032-7698

SIGNATURE 

DATE 6/29/00

Burden Hour Statement: This form is estimated to take 3 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231

Methods and Apparatus for Managing a Buffer of Events in the Background

*by inventors Mario Nemirovsky, Naren Sankar,
Adolfo Nemirovsky and Enric Musoll*

5

Field of the Invention

10 The present invention is in the area of integrated circuit microprocessors, and pertains in particular to ordering the activity of a processor in response to receipt and storage of data to be processed.

15

Cross-Reference to Related Documents

 The present patent application is a Continuation-In-Part of co-pending patent application attorney docket number P3814 entitled "Methods and Apparatus for Background Memory Management" filed on 06/23/2000. The prior application is incorporated herein in its entirety by reference.

20

Background of the Invention

25 Microprocessors, as is well-known in the art, are integrated circuit (IC) devices that are enabled to execute code sequences which may be generalized as software. In the execution most microprocessors are capable of both logic and arithmetic operations, and typically modern microprocessors have on-chip resources (functional units) for such processing.

Microprocessors in their execution of software strings typically operate on data that is stored in memory. This data needs to be brought into the memory before the processing is done, and sometimes needs to be sent out to a device that needs it after its processing.

5 There are in the state-of-the-art two well-known mechanisms to bring data into the memory and send it out to a device when necessary. One mechanism is loading and storing the data through a sequence of Input/Output (I/O) instructions. The other is through a direct-memory access device (DMA).

10 In the case of a sequence of *I/O* instructions, the processor spends significant resources in explicitly moving data in and out of the memory. In the case of a DMA system, the processor programs an external hardware circuitry to perform the data transferring. The DMA circuitry performs all of the required memory accesses to perform the data transfer to and from the
15 memory, and sends an acknowledgement to the processor when the transfer is completed.

 In both cases of memory management in the art the processor has to explicitly perform the management of the memory, that is, to decide whether the desired data structure fits into the available memory space or
20 does not, and where in the memory to store the data. To make such decisions the processor needs to keep track of the regions of memory wherein useful data is stored, and regions that are free (available for data storage). Once that data is processed, and sent out to another device or location, the region of memory formerly associated with the data is free to be
25 used again by new data to be brought into memory. If a data structure fits into the available memory, the processor needs to decide where the data structure will be stored. Also, depending on the requirements of the processing, the data structure can be stored either consecutively, in which

case the data structure must occupy one of the empty regions of memory; or non-consecutively, wherein the data structure may be partitioned into pieces, and the pieces are then stored into two or more empty regions of memory.

5 An advantage of consecutively storing a data structure into memory is that the accessing of this data becomes easier, since only a pointer to the beginning of the data is needed to access all the data.

10 When data is not consecutively stored into the memory, access to the data becomes more difficult because the processor needs to determine the explicit locations of the specific bytes it needs. This can be done either in software (i.e. the processor will spend its resources to do this task) or in hardware (using a special circuitry). A drawback of consecutively storing the data into memory is that memory fragmentation occurs. Memory
15 fragmentation happens when the available chunks of memory are smaller than the data structure that needs to be stored, but the addition of the space of the available chunks is larger than the space needed by the data structure. Thus, even though enough space exists in the memory to store the data structure, it cannot be consecutively stored. This drawback does not exist if the data structure is allowed to be non-consecutively stored.

20 Still, a smart mechanism is needed to generate the lowest number of small regions, since the larger the number of small regions that are used by a data structure, the more complex the access to the data becomes (more specific regions need to be tracked) regardless of whether the access is managed in software or hardware as explained above.

25 A related problem in processing data is in the establishment of an order of processing in response to an order of receiving data to be processed. In many cases, data may be received and stored faster than a processor can process the data, and there is often good reason for processing data in an order different from the order in which the data is received. In the

current art, for a processor to take priorities into account in the order in which it processes data, the processor has to expend resources on checking the nature of the data (priorities) and in re-ordering the sequence in which it will process the data.

5 What is clearly needed is a background system for tracking data receipt and storage for a processor system, and for ordering events for the processor.

10 **Summary of the Invention**

 In a preferred embodiment of the present invention, a background event buffer manager (BEBM) for ordering and accounting for events in a data processing system having a processor is provided, the BEBM
15 comprising a port for receiving event identifications (IDs) from a device; a queuing function enabled for queuing event IDs received; and a notification function for notifying the processor of queued event IDs. The BEBM handles all event ordering and accounting for the processor.

 In preferred embodiments the queuing function queues event IDs by
20 type, and by event priority within type queues, and also associates an acknowledgment (ack) with each event. The ack may have multiple states, and acks for events first queued are set in a "processor unaware" state. Also, after notification to the processor the ack state for the queued event is changed to a state of "processor aware".

25 In some embodiments there is further a function for receiving notification from the processor of processing completed on an event, and wherein, upon receiving such a notification the state of the ack for the event is changed to "ready". There may further be a function for sending acks in

ready state back to the device that originally sent the event associated with the ack, and for buffering the process of sending the acks.

In preferred embodiments the processor is notified of events in order of type priority first, and then by event priority within type. Also in
5 preferred embodiments, the events may be arrival of packets to be processed in a network packet router.

In another aspect of the invention a data processing system is provided, comprising a processor; a memory coupled to the processor; and a background event buffer manager BEBM coupled to the processor, the
10 BEBM including a port for receiving event identifications (IDs) from a device, a queuing function enabled for queuing event IDs received, and a notification function for notifying the processor of queued event IDs. The system is characterized in that the BEBM handles all event ordering and accounting for the processor.

In the data processing system the queuing function, in preferred
15 embodiments, queues event IDs by type, and by event priority within type queues, and also associates an acknowledgment (ack) with each event.

In some embodiments the ack may have multiple states, and acks for events first queued are set in a "processor unaware" state. After notification
20 to the processor the ack state for the queued event may be changed to a state of "processor aware". In some embodiments there is also a function for receiving notification from the processor of processing completed on an event, and wherein, upon receiving such a notification the state of the ack for the event is changed to "ready". Also in some embodiments there is a
25 function for sending acks in ready state back to the device that originally sent the event associated with the ack, and for buffering the process of sending the acks.

In preferred embodiments of the data processing system the processor is notified of events in order of type priority first, and then by event priority within type. In some embodiments the events are arrival of packets to be processed in a network packet router, and the system is a packet processing engine.

In yet another aspect of the invention a network packet router is provided, comprising an input/output (I/O) device for receiving and sending packets on the network, a processor, a memory coupled to the processor, and a background event buffer manager BEBM coupled to the processor, the BEBM including a port for receiving event identifications (IDs) of arriving packets from the I/O device, a queuing function enabled for queuing packet IDs received, and a notification function for notifying the processor of queued packet IDs for processing. The router is characterized in that the BEBM handles all event ordering and accounting for the processor.

In preferred embodiments of the router the queuing function queues event IDs by type, and by event priority within type queues, and also associates an acknowledgment (ack) with each event. The ack may have multiple states, and acks for events first queued are set in a "processor unaware" state. After notification to the processor the ack state for the queued event may be changed to a state of "processor aware".

In some embodiments of the router there is a function for receiving notification from the processor of processing completed on an event, and wherein, upon receiving such a notification the state of the ack for the event is changed to "ready". There may further be a function for sending acks in ready state back to the device that originally sent the event associated with the ack, and for buffering the process of sending the acks. In preferred embodiments of the router the processor is notified of events in order of type priority first, and then by event priority within type.

In yet another aspect of the invention a method for ordering and accounting for events in a data processing system having a processor is provided, the method comprising steps of (a) generating event identifications (IDs) by a device; (b) sending the event IDs to a background event buffering manager (BEBM) by the device; (c) queuing event IDs received by the
5 BEBM; and (d) notifying the processor by the BEBM of events queued for processing, such that the BEBM handles all event ordering and accounting for the processor.

In preferred embodiments of the method, in step (c), event IDs are
10 queued by type, and by event priority within type queues, and an acknowledgment (ack) is associated with each event. Also in preferred embodiments the ack may have multiple states, and acks for events first queued are set in a "processor unaware" state. After notification to the processor the ack state for the queued event may be changed to a state of
15 "processor aware".

In some embodiments of the method there is a further step for receiving, at the BEBM, notification from the processor of processing completed on an event, and wherein, upon receiving such a notification the state of the ack for the event is changed to "ready". Also in some
20 embodiments there is a step for sending acks in ready state back to the device that originally sent the event associated with the ack, and for buffering the process of sending the acks. The processor may be notified of events in order of type priority first, and then by event priority within type.

In embodiments of the invention taught in enabling detail below, for
25 the first time a background event buffer manager is provided for processing systems, so events for processing may be queued by priority ahead of the processor, and all accounting for events may be handled with a minimum of activity by the processor itself.

Brief Description of the Drawing Figures

Fig. 1 is a simplified diagram of memory management by direct I/O processing in the prior art.

Fig. 2 is a simplified diagram of memory management by direct memory access in the prior art.

Fig. 3 is a diagram of memory management by a Background Memory Manager in a preferred embodiment of the present invention.

Fig. 4 is a diagram of ordering of events for a processor by a Background Event Manager (BEBM) according to an embodiment of the present invention.

Fig. 5 is a high-level diagram of queues and functions for a background event buffer manager according to an embodiment of the present invention.

Description of the Preferred Embodiments

Fig. 1 is a simplified diagram of memory management in a system 104 comprising a processor 100 and a memory 102 in communication with a device 106. In this example it is necessary to bring data from device 106 into memory 102 for processing, and sometimes to transmit processed data from memory 102 to device 106, if necessary. Management in this prior art example is by processor 100, which sends I/O commands to and receives

responses and/or interrupts from device 106 via path 108 to manage movement of data between device 106 and memory 102 by path 110. The processor has to determine whether a data structure can fit into available space in memory, and has to decide where in the memory to store incoming data structures. Processor 100 has to fully map and track memory blocks into and out of memory 102, and retrieves data for processing and stores results, when necessary, back to memory 102 via path 114. This memory management by I/O commands is very slow and cumbersome and uses processor resources quite liberally.

Fig. 2 is a simplified diagram of a processor system 200 in the prior art comprising a processor 100, a memory 102 and a direct memory access (DMA) device 202. This is the second of two systems by which data, in the conventional art, is brought into a system, processed, and sent out again, the first of which is by I/O operations as described just above. System 200 comprises a DMA device 202 which has built-in intelligence, which may be programmed by processor 100, for managing data transfers to and from memory 102. DMA device 202 is capable of compatible communication with external device 106, and of moving blocks of data between device 102 and 106, bi-directionally. The actual data transfers are handled by DMA device 202 transparently to processor 100, but processor 100 must still perform the memory mapping tasks, to know which regions of memory are occupied with data that must not be corrupted, and which regions are free to be occupied (overwritten) by new data.

In the system of Fig. 2 DMA processor 100 programs DMA device 202. This control communication takes place over path 204. DMA device 202 retrieves and transmits data to and from device 106 by path 208, and handles data transfers between memory 102 and processor 100 over paths 204 and 206.

In these descriptions of prior art the skilled artisan will recognize that paths 204, 206 and 208 are virtual representations, and that actual data transmission may be by various physical means known in the art, such as by parallel and serial bus structures operated by bus managers and the like, the bus structures interconnecting the elements and devices shown.

Fig. 3 is a schematic diagram of a system 300 including a Background Memory Manager (BMM) 302 according to an embodiment of the present invention. BMM 302 a hardware mechanism enabled to manage the memory in the background, i.e. with no intervention of the processor to decide where the data structure will be stored in the memory. Thus, the processor can utilize its resources for tasks other than to manage the memory.

The present invention in several embodiments is applicable in a general way to many computing process and apparatus. For example, in a preferred embodiment the invention is applicable and advantageous in the processing of data packets at network nodes, such as in packet routers in the Internet. The packet processing example is used below as a specific example of practice of the present invention to specifically describe apparatus, connectivity and functionality.

In the embodiment of a packet router, device 106 represents input/output apparatus and temporary storage of packets received from and transmitted on a network over path 308. The network in one preferred embodiment is the well-known Internet network. Packets received from the Internet in this example are retrieved from device 106 by BMM 302, which also determines whether packets can fit into available regions in memory and exactly where to store each packet, and stores the packets in memory 102, where they are available to processor 100 for processing. Processor 100 places results of processing back in memory 102, where the processed

packets are retrieved, if necessary, by BMM on path 312 and sent back out through device 106.

In the embodiment of Fig. 3 BMM 302 comprises a DMA 202 and also a memory state map 304. BMM 302 also comprises an interrupt handler in a preferred embodiment, and device 106 interrupts BMM 302 when a packet is received. When a packet is received, using DMA 202 and state map 304, the BMM performs the following tasks:

1. Decides whether a data structure fits into the memory. Whether the structure fits into memory, then, is a function of the size of the data packet and the present state of map 304, which indicates those regions of memory 102 that are available for new data to be stored.

2. If the incoming packet in step 1 above fits into memory, the BMM determines an optimal storage position. It was described above that there are advantages in sequential storage. Because of this, the BMM in a preferred embodiment stores packets into memory 102 in a manner to create a small number of large available regions, rather than a larger number of smaller available regions.

3. BMM 302 notifies processor 100 on path 310 when enough of the packet is stored, so that the processor can begin to perform the desired processing. An identifier for this structure is created and provided to the processor. The identifier communicates at a minimum the starting address of the packet in memory, and in some cases includes additional information.

4. BMM updates map 304 for all changes in the topology of the memory. This updating can be done in any of several ways, such as periodically, or every time a unit in memory is changed.

5 5. When processing is complete on a packet the BMM has stored in memory 102, the processor notifies BMM 302, which then transfers the processed data back to device 106. This is for the particular example of a packet processing task. In some other embodiments data may be read out of memory 102 by MM 302 and sent to different devices, or even discarded. In
10 notifying the BMM of processed data, the processor used the data structure identifier previously sent by the BMM upon storage of the data in memory 102.

15 6. The BMM updates map 304 again, and every time it causes a change in the state of memory 102. Specifically the BMM de-allocates the region or regions of memory previously allocated to the data structure and sets them as available for storage of other data structures, in this case packets.

In another aspect of the invention methods and apparatus are provided for ordering events for a processor other than the order in which
20 data might be received to be processed, and without expenditure of significant processor resources.

In the teachings above relative to background memory management an example of packet routing in networks such as the Internet was used extensively. The same example of Internet packet traffic is particularly
25 useful in the present aspect of event managing for a processor, and is therefore continued in the present teaching.

Fig. 4 is a system diagram illustrating a system 407 using a Background Event Buffer Manager (BEBM) 401 according to an

embodiment of the present invention. In this embodiment BEBM 401 works in conjunction with BMM 302 described in enabling detail above. Further, for purposes of illustration and description, it is assumed that system 407 in Fig. 4 is a system operating in a packet router, and the network to which device 106 connects by link 308 is the well-known Internet network.

In a communication session established over the Internet between any two sites there will be an exchange of a large number of packets. For the purpose of the present discussion we need to consider only flow of packets in one direction, for which we may select either of the two sites as the source and the other as the destination. In this example packets are generated by the source, and received at the destination. It is important that the packets be received at the destination in the same order as they are generated and transmitted at the source, and, if the source and destination machines were the only two machines involved with the packet flow, and all packets in the flow were to travel by the same path, there would be no problem. Packets would necessarily arrive in the order sent.

Unfortunately packets from a source to a destination may flow through a number of machines and systems on the way from source to destination, and there are numerous opportunities for packets to get disordered. Moreover, the machines handling packets at many places in the Internet are dealing with large numbers of sources and destinations, and therefore with a large number of separate packet flows, which are termed microflows in the art. It will be apparent to the skilled artisan that packets from many different microflows may be handled by a single router, and the packets may well be intermixed while the packets for each separate microflow are still in order. That is, packets from one microflow may be processed, then packets from a second and third microflow, and then more

packets from the first microflow, while if only packets from one microflow are considered the flow is sequential and orderly.

The problems that can occur if microflows are allowed to be disordered are quite obvious. If a particular microflow is for an Internet telephone conversation, for example, and the flow gets out-of-order the audio rendering may be seriously affected. Systems for Internet communication are, of course, provided in the art with re-ordering systems for detecting disordered microflows, and re-ordering the packets at the destination, but such systems require a considerable expenditure of processing resources, and, in some cases, packets may be lost or discarded.

It will also be apparent to the skilled artisan that packets from a source to a destination typically pass through and are processed by a number of different machines along the way from source to destination. System 407 illustrated in Fig. 4 is meant to represent one such system through which packets may pass and be processed, and such systems include source and end nodes as well. It should be apparent, then, that, since packets originate in a specific order at a source site, they will highly likely be received in that order at a first destination, and that therefore a reasonable goal at any router will be to always process and retransmit packets from a router in the same order that the packets were received for a specific microflow.

Referring now to Fig. 3, which is the system of Fig. 4 without the BEBM of the present invention, it was described above that Background Memory Manager (BMM) 302 handles all memory state accounting and moves all data, in this example packets, into and out of memory 102. In that procedure the processor is informed by the BMM via path 310, as an interrupt, when a packet from device 106 is stored in memory 102. When the processor has processed a packet, device 106 must be notified, which is done through BMM 302.

Now, it is well known that packets are not necessarily received in a steady flow, but may be received in bursts. Still, BMM 302 in the case of the system of Fig. 3, or processor 100 in the case of most prior art systems, is interrupted for each packet. It may be that when the processor is interrupted for packets arriving that the processor is busy on other tasks. Interrupts may arrive in bursts much faster than the processor can handle them. In this case the processor has to keep track of all of the events and order the processing of all events.

In a somewhat more general sense the process just described, sans BEBM, can be described as follows:

In some applications a processor needs to perform some kind of processing when an event generated by a device occurs, and it has to notify that device when the processing of the event is completed (henceforth this notification is named *ack*, for acknowledge).

An event e generated by a device may have associated a type of a priority p (henceforth named *type priority*). Within a type priority, events can have different priorities q (henceforth named *event priority*). The device may generate the events in any order. However, it may impose some restrictions on the ordering of the corresponding acks. A reason why this can happen is because the device may not know the type priority nor the event priority of the event it generates, and therefore it relies on the processing of the events to figure out the type and/or event priorities. Thus, it may request the acks of the highest priority events to be received first.

More specifically, let $Gen(e)$ be the time when event e was generated by the device; $Gen(e1) < Gen(e2)$ indicates that event $e1$ was generated before than event $e2$. Let $Ack(e)$ be the time when the ack is generated for event e by the processor; $Ack(e1) < Ack(e2)$ indicates that the ack for event

$e1$ was generated before the ack for event $e2$. Let $e(p)$ and $e(q)$ be the type priority and event priority, respectively, of event e .

The following are examples of restrictions that the device can impose on the ordering of the acks generated by the processor. The
5 device might request, for example, that

(a) $Ack(e1) < Ack(e2)$ when $Gen(e1) < Gen(e2)$

Acks are generated in the same order that the corresponding events
10 occurred, independently on the type and event priority of the events.

(b) $Gen(e1) < Gen(e2)$ AND $e1(p) = e2(p)$

Acks for the events of the same type priority are generated in the
15 same order that the events where generated;

(c) $e1(p) > e2(p)$

Acks for the events with highest event priority (that the processor is
20 currently aware of) are generated first.

(d) $e1(q) > e2(q)$.

Acks for the events of the highest type priority (of which the
25 processor is currently aware) are generated first.

(e) $e1(q) > e2(q)$ AND $e1(p) > e2(p)$

Acks for the events with highest event priority in the highest type priority (of which the processor is currently aware) are generated first.

In any case, the goal of the processor is to generate the acks as soon as possible to increase the throughput of processed events. The processor
5 can dedicate its resources to guarantee the generation of acks following the restrictions mentioned above. However, the amount of time the processor dedicates to this task can be significant, thus diminishing the performance of the processor on the processing of the events. Moreover, depending on how frequent these events occur, and the amount of processing that each event
10 takes, the processor will not be able to start processing them at the time they occur. Therefore, the processor will need to buffer the events and process them later on.

The skilled artisan will surely recognize that the ordering of and accounting for events, as described herein, is a considerable and significant
15 processor load.

In preferred embodiments of the present invention, managing of the ordering of the acks is implemented in hardware and is accomplished in the background (i.e. while the processor is performing the processing of other events).

20 The system at the heart of embodiments of the present invention is called by the inventors a Background Event Buffer Manager (BEBM), and this is element 401 in Fig. 4. Note that in Fig. 4 the BEBM is implemented in system 407 with the inventive BMM 302, which is described in enabling detail above. It is not a restriction on the implementation of BEBM 401 that
25 it only perform with a BMM in a system, but it is a convenience adding further advantages in a processing system.

The BEBM performs the following tasks:

1. The BEBM buffers, completely in the background, all incoming events. If proper information is available (like the type priority and event priority described above), the buffering of an event will be done so that its ack happens as soon as possible under the restrictions in task 4, below. In other words, the BEBM will take advantage of any information about the event that the device can generate. In some cases the device may generate events without an associated priority.

2. The BEBM notifies the processor about any event that has been buffered and for which the processor does not know of its existence. Similarly as in task 1, if the proper information is available (like the type priority and event priority), these notifications will be generated so that the ack happens as soon as possible under the restrictions in task 4. The processor will typically follow the priorities, if notified, but may, under certain conditions override or change the priority of an event.

3. The BEBM updates the status (further description below) of the ack for a particular event based on the result of the processing of the event by the processor;

4. The BEBM guarantees any of the restrictions imposed on the generation of the acks with minimal intervention of the processor.

When an event is buffered in the BEBM (task 1), its corresponding ack, also buffered with the event, is in the *processor-not-aware* state, meaning that the processor still has no knowledge of this event and, therefore, it still has to start processing it. When the event is presented to the

processor (task 2), its ack state transitions into *processor-aware*, meaning that the processor has started the processing of the event but it still has not finished. When the processor finishes this processing, it notifies the BEBM about this fact and the state of the ack becomes *ready*.

5 At this point, the ack becomes a candidate to be sent out to the device. When the ack state becomes *processor-aware*, the associated information to the event (type priority and event priority) may be provided to the processor or not, depending on whether the device that generated the event also generated this information or not. The processor can figure out
10 this information during the processing of the event, and override the information sent by the device, if needed. This information can potentially be sent to the BEBM through some communication mechanism. The BEBM records this information, which is used to guarantee task 4.

 In case the processor does not communicate this information to the
15 BEBM, the original information provided by the device, if any, or some default information will be used to guarantee task 4. In any case, the processor always needs to communicate when the processing of an event has completed.

 When the processor finishes processing an event the state of the ack
20 associated with the event ID is changed to "ready", as previously described. The BEBM also buffers the transmission of acks back to the device that generated the events, because the device may be busy with other tasks. As the device becomes capable of processing the acks, the BEBM sends them to the device.

25 Fig. 5 is a high-level diagram of a BEBM according to an embodiment of the present invention. The squares represent the buffering and queuing part of the BEBM and the ellipses represent the different tasks 1-4 described above.

Ideally, the buffering function of the BEBM is divided into as many queues (blocks) as different types of priorities exist, and each block has as many entries as needed to buffer all the events that might be encountered, which may be determined by the nature of the hardware and application. In the instant example three queues are shown, labeled 1, 2 and P, to represent an indeterminate number. In a particular implementation of the BEBM, however, the amount of buffering will be limited and, therefore several priority types might share the same block, and/or a block might get full, so no more events will be accepted. This limitation will affect how fast the acks are generated but it will not affect task 4.

In the context of the primary example in this specification, that of a packet processing engine, the data structures provided by device 106 are network packets, and the events are therefore the fact of receipt of new packets to be processed. There will typically be packets of different types, which may have type priorities, and within types there may also be event priorities. The BEBM therefore maintains queues for different packet types, and offers the queues to the processor by priority; and also orders the packets in each type queue by priority (which may be simply the order received).

In the instant example, referring now to Fig. 4, the events come from the BMM to task 1, the processor is notified by event id (task 2) by the BEBM, and the processor, having processed a packet sends a command to update the ack associated with the event id. The BEBM then sends the updated ack (task 4) to the BMM, which performs necessary memory transfers and memory state updates.

It will be apparent to the skilled artisan that there may be many alterations in the embodiments described above without departing from the spirit and scope of the present invention. For example, a specific case of

operations in a data packet router has been illustrated. This is a single instance of a system wherein the invention may provide significant advantages. There are many other systems and processes that will benefit as well. Further, there are a number of ways a BEBM and BMM may be implemented, either alone or together, to perform the functionality described above, and there are many systems incorporating many different kinds of processors that might benefit. The present inventors are particularly interested in a system wherein a dynamic multi-streaming processor performs the functions of processor 100. For these and other reasons the invention should be limited only by the scope of the claims below.

5

10

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
220

What is claimed is:

1. A background event buffer manager (BEBM) for ordering and accounting for events in a data processing system having a processor, the
5 BEBM comprising:

a port for receiving event identifications (IDs) from a device;
a queuing function enabled for queuing event IDs received; and
a notification function for notifying the processor of queued event
IDs;

10 characterized in that the BEBM handles all event ordering and accounting for the processor.

2. The BEBM of claim 1 wherein the queuing function queues event IDs by type, and by event priority within type queues, and also associates an
15 acknowledgment (ack) with each event.

3. The BEBM of claim 2 wherein the ack may have multiple states, and acks for events first queued are set in a "processor unaware" state.

20 4. The BEBM of claim 3 wherein, after notification to the processor the ack state for the queued event is changed to a state of "processor aware".

5. The BEBM of claim 4 comprising a function for receiving notification from the processor of processing completed on an event, and wherein, upon
25 receiving such a notification the state of the ack for the event is changed to "ready".

6. The BEBM of claim 5 further comprising a function for sending acks in ready state back to the device that originally sent the event associated with the ack, and buffering the process of sending the acks.

5 7. The BEBM of claim 2 wherein the processor is notified of events in order of type priority first, and then by event priority within type.

8. The BEBM of claim 1 wherein the events are arrival of packets to be processed in a network packet router.

10

9. A data processing system, comprising:

 a processor;

 a memory coupled to the processor; and

 a background event buffer manager BEBM coupled to the processor,

15 the BEBM including a port for receiving event identifications (IDs) from a device, a queuing function enabled for queuing event IDs received, and a notification function for notifying the processor of queued event IDs;

 characterized in that the BEBM handles all event ordering and accounting for the processor.

20

10. The data processing system of claim 9 wherein the queuing function queues event IDs by type, and by event priority within type queues, and also associates an acknowledgment (ack) with each event.

25 11. The data processing system of claim 10 wherein the ack may have multiple states, and acks for events first queued are set in a "processor unaware" state.

12. The data processing system of claim 11 wherein, after notification to the processor the ack state for the queued event is changed to a state of "processor aware".

5 13. The data processing system of claim 12 comprising a function for receiving notification from the processor of processing completed on an event, and wherein, upon receiving such a notification the state of the ack for the event is changed to "ready".

10 14. The data processing system of claim 13 further comprising a function for sending acks in ready state back to the device that originally sent the event associated with the ack, and for buffering the sending process.

15 15. The data processing system of claim 10 wherein the processor is notified of events in order of type priority first, and then by event priority within type.

20 16. The data processing system of claim 9 wherein the events are arrival of packets to be processed in a network packet router, and the system is a packet processing engine.

17. A network packet router, comprising:
an input/output (I/O) device for receiving and sending packets on the network;
25 a processor;
a memory coupled to the processor; and
a background event buffer manager BEBM coupled to the processor,
the BEBM including a port for receiving event identifications (IDs) of

arriving packets from the I/O device, a queuing function enabled for queuing packet IDs received, and a notification function for notifying the processor of queued packet IDs for processing;

5 characterized in that the BEBM handles all event ordering and accounting for the processor.

10 18. The network packet router of claim 17 wherein the queuing function queues event IDs by type, and by event priority within type queues, and also associates an acknowledgment (ack) with each event.

19. The network packet router of claim 18 wherein the ack may have multiple states, and acks for events first queued are set in a "processor unaware" state.

15 20. The network packet router of claim 19 wherein, after notification to the processor the ack state for the queued event is changed to a state of "processor aware".

20 21. The network packet router of claim 20 comprising a function for receiving notification from the processor of processing completed on an event, and wherein, upon receiving such a notification the state of the ack for the event is changed to "ready".

25 22. The network packet router of claim 21 further comprising a function for sending acks in ready state back to the device that originally sent the event associated with the ack, and for buffering the sending process.

22. The network packet router of claim 18 wherein the processor is notified of events in order of type priority first, and then by event priority within type.

5 23. The network packet router of claim 17 wherein the events are arrival of packets to be processed in a network packet router, and the system is a packet processing engine.

10 24. A method for ordering and accounting for events in a data processing system having a processor, the method comprising steps of:

- (a) generating event identifications (IDs) by a device;
- (b) sending the event IDs to a background event buffering manager (BEBM) by the device;
- (c) queuing event IDs received by the BEBM; and
- 15 (d) notifying the processor by the BEBM of events queued for processing, such that the BEBM handles all event ordering and accounting for the processor.

20 25. The method of claim 24 wherein, in step (c), event IDs are queued by type, and by event priority within type queues, and an acknowledgment (ack) is associated with each event.

26. The method of claim 25 wherein the ack may have multiple states, and acks for events first queued are set in a "processor unaware" state.

25 27. The method of claim 26 wherein, after notification to the processor the ack state for the queued event is changed to a state of "processor aware".

28. The method of claim 27 comprising a further step for receiving, at the BEBM, notification from the processor of processing completed on an event, and wherein, upon receiving such a notification the state of the ack for the event is changed to "ready".

5

29. The method of claim 28 further comprising a step for sending acks in ready state back to the device that originally sent the event associated with the ack, and for buffering the sending process.

10

30. The method of claim 25 wherein the processor is notified of events in order of type priority first, and then by event priority within type.

15

31. The method of claim 24 wherein the events are arrival of packets to be processed in a network packet router, and the system is a packet processing engine.

Abstract of the Disclosure

5 A background event buffer manager (BEBM) for ordering and
accounting for events in a data processing system having a processor
includes a port for receiving event identifications (IDs) from a device, a
queuing function enabled for queuing event IDs received, and a notification
function for notifying the processor of queued event IDs. The
10 BEBM handles all event ordering and accounting for the processor. The
BEBM in preferred embodiments queues events by type with priority and by
priority within type, and also handles sending acknowledgement to the
device when processing on each event is concluded, and buffers the
acknowledgement process. In particular embodiments the apparatus and
15 method is taught as a packet processing router engine.

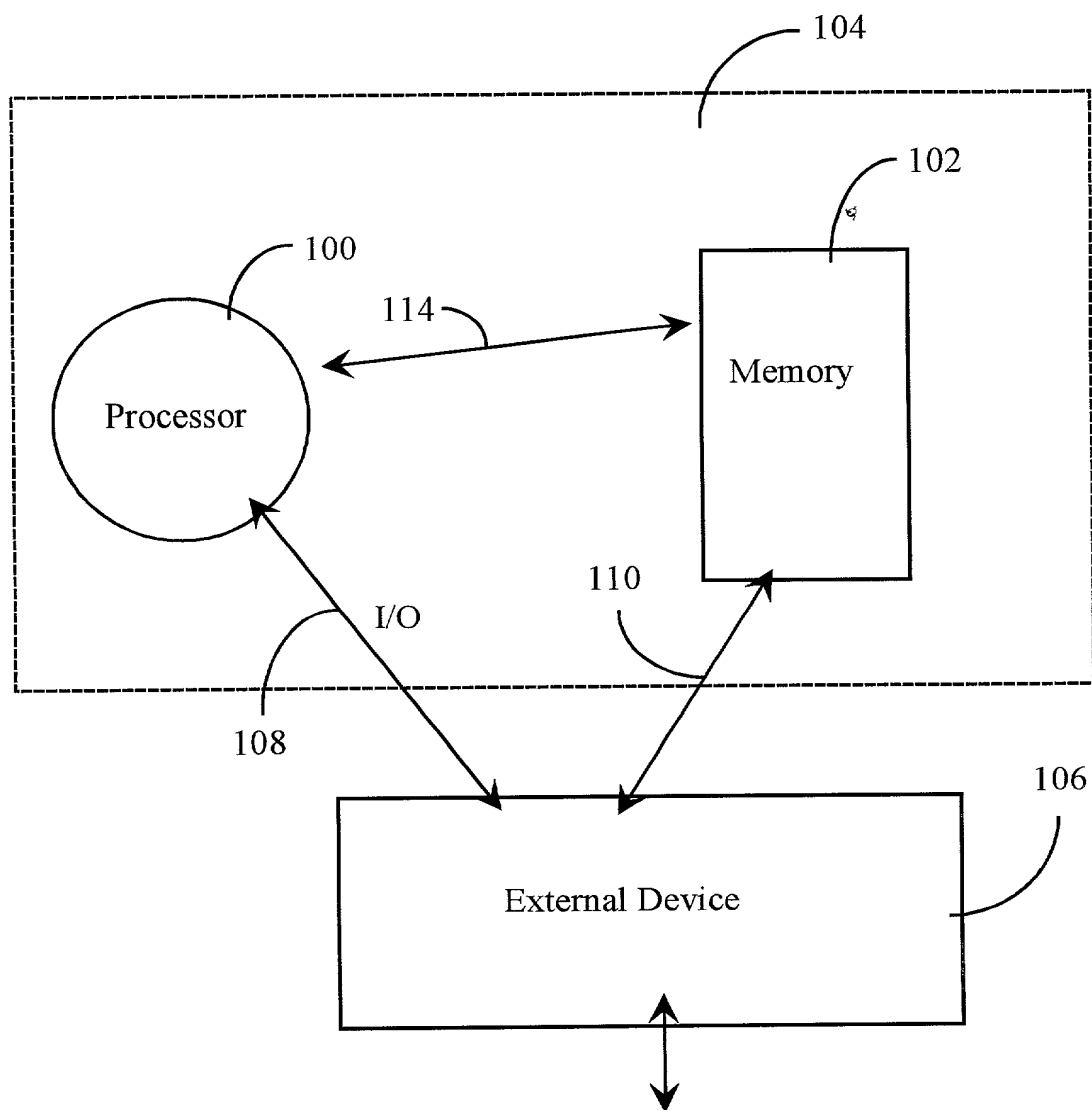


Fig. 1 (Prior Art)

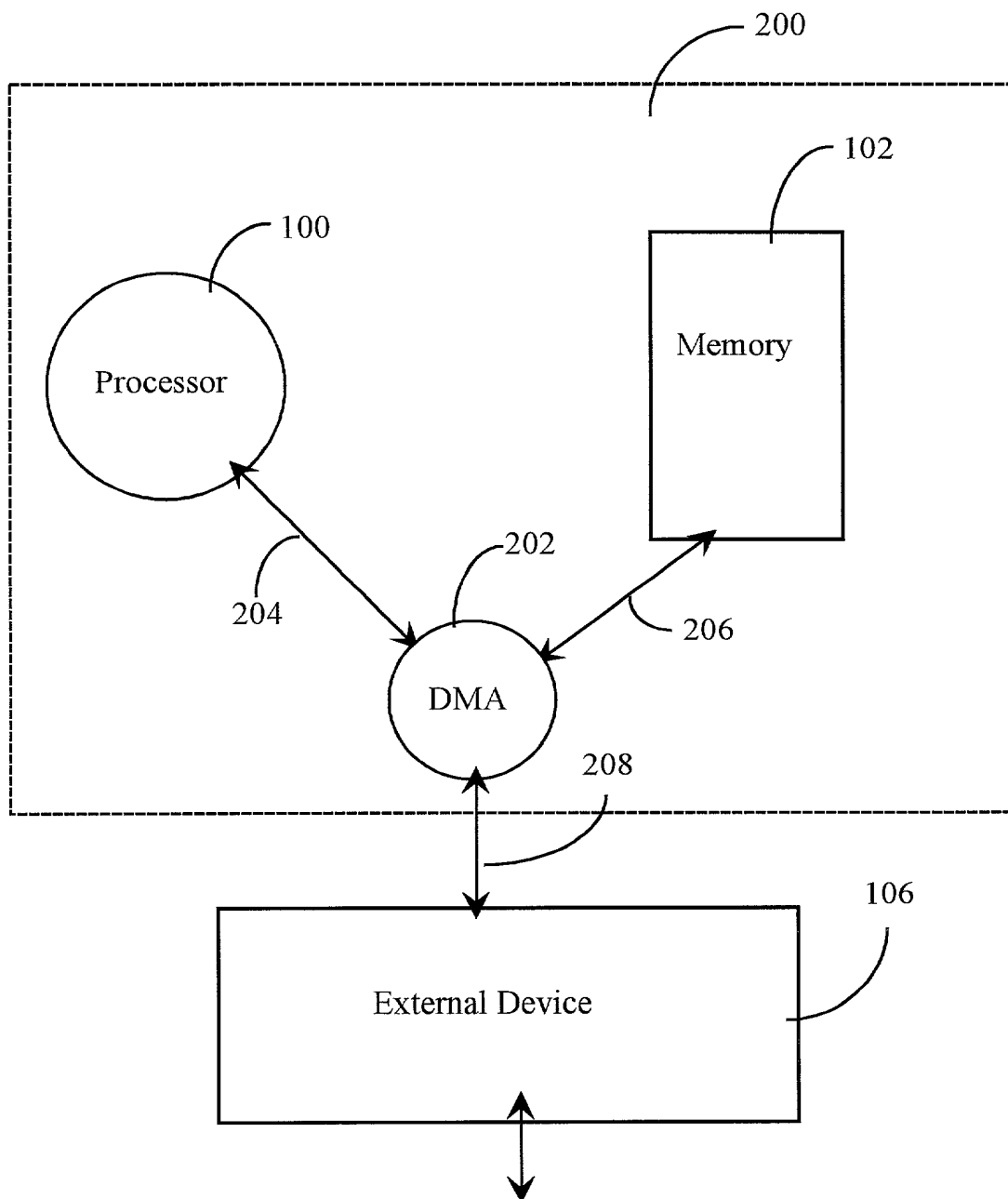


Fig. 2 (Prior Art)

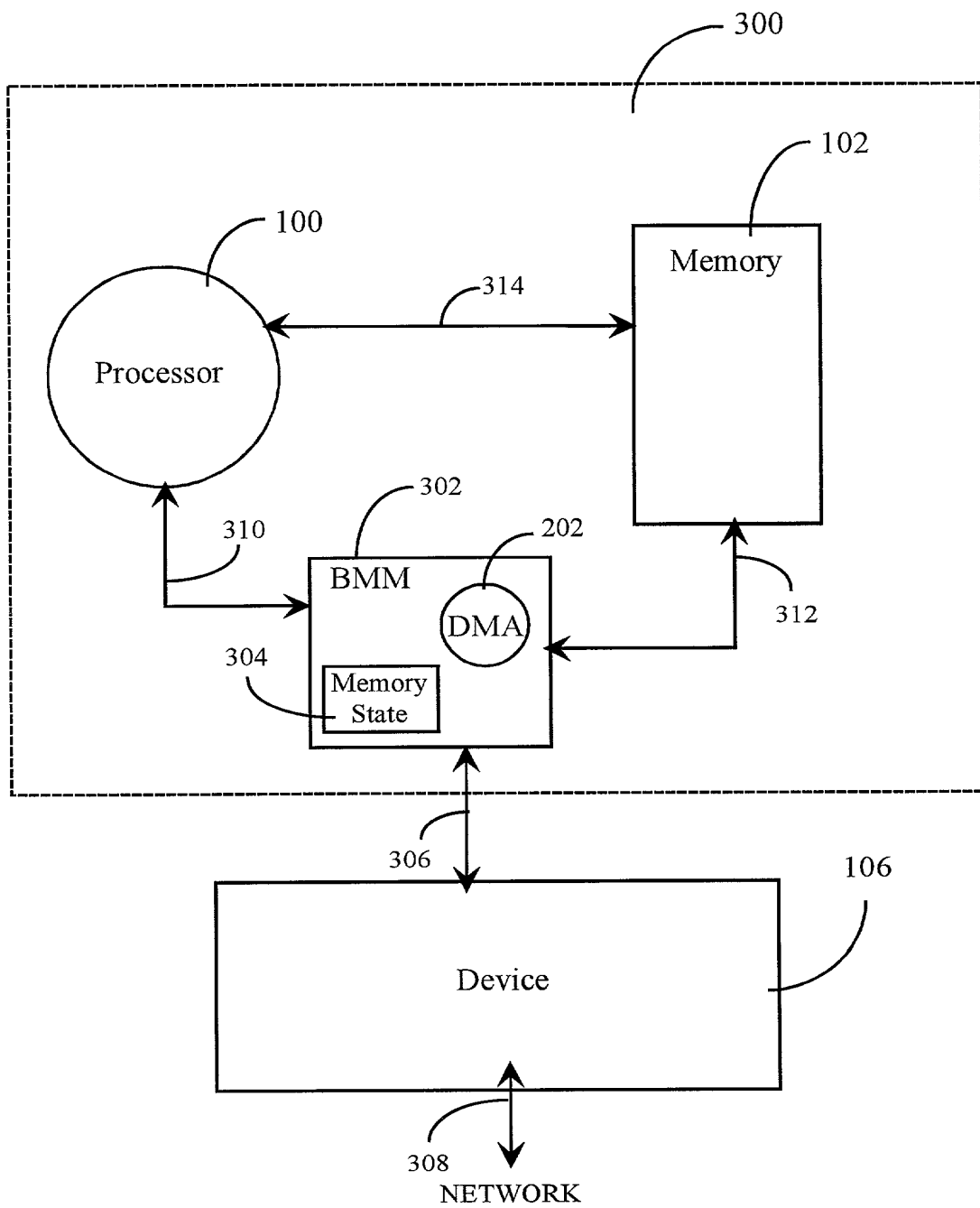


Fig. 3

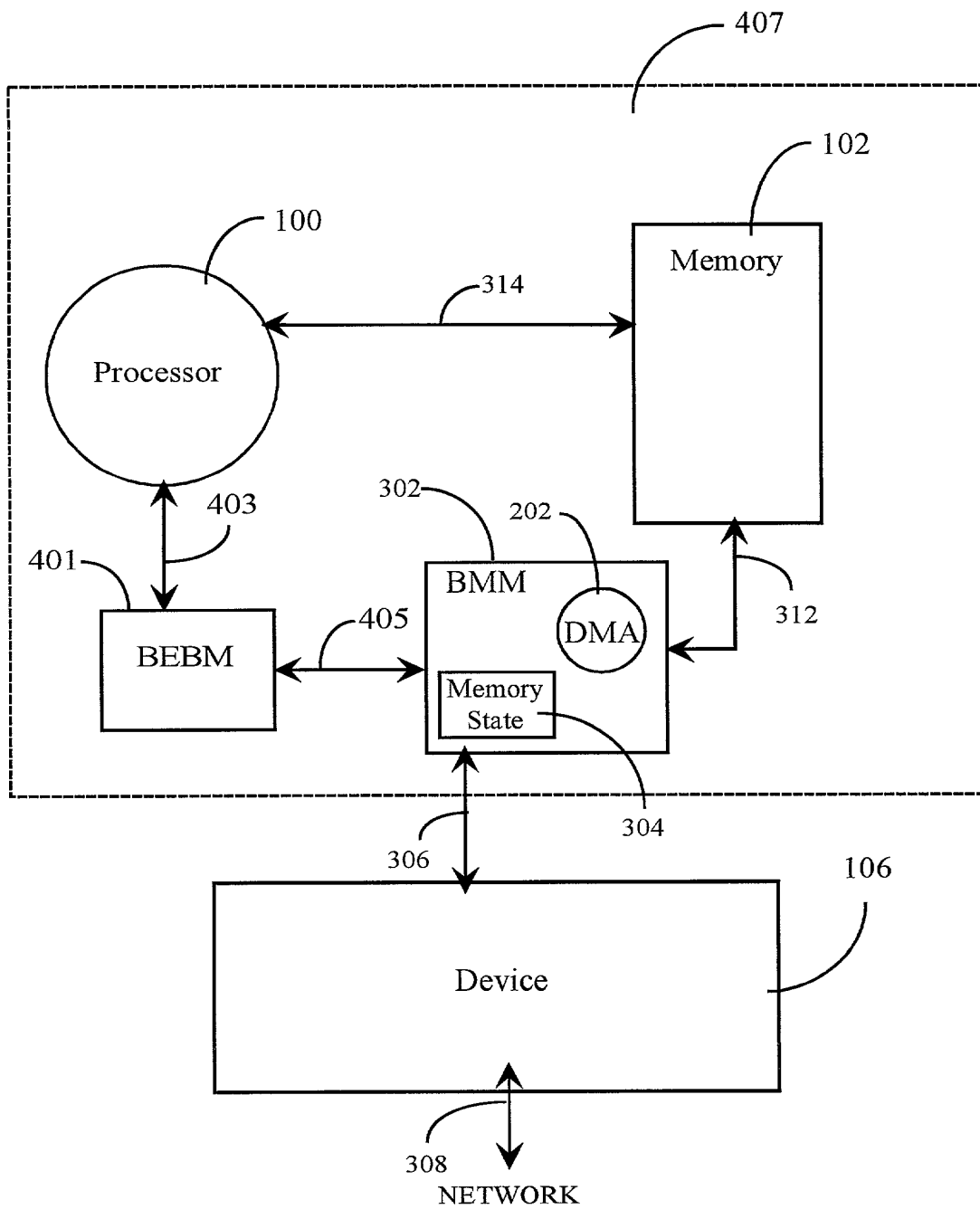


Fig. 4

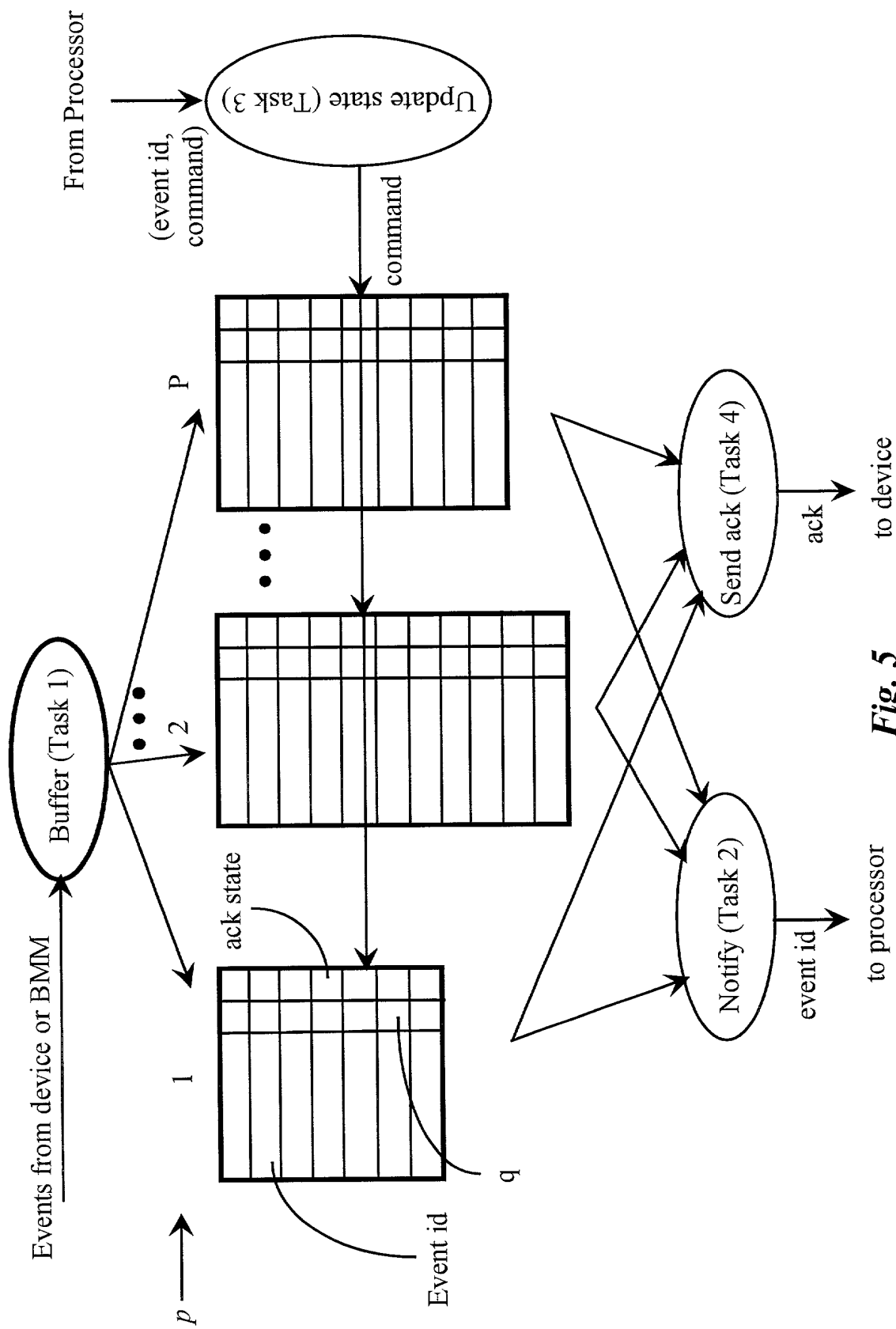


Fig. 5

3815

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

ATTORNEY DOCKET NO. P3815

As a below named inventor, I hereby declare that: My residence, post office address and citizenship are as stated below next to my name. I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled: Methods and Apparatus for Managing a Buffer of Events in the Background

the specification of which (check one) ☒ is attached hereto.

☐ was filed on: _____
☐ Application Serial No. _____
☐ and was amended on _____
 (If applicable)

I hereby state that I have reviewed and understood the contents of the above-identified specification, including the claims, as amended by any amendment referred to above. I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, s 1.56 (a). In the case that the present application is a continuation-in-part application, I further acknowledge the duty to disclose material information as defined in 37 CFR s 1.56(a) which became available between the filing date of the prior application and the filing date of the present application. I hereby claim foreign priority benefits under Title 35, United States Code s 119 of any foreign applications for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)

(Number)	(Country)	(Day/Month/Year Filed)
----------	-----------	------------------------

(Number)	(Country)	(Day/Month/Year Filed)
----------	-----------	------------------------

I hereby claim the benefit under Title 35, United States Code, s 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, s 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, s 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application.

(Application Serial No.): _____	(Filing Date): _____	(Status): _____
(Application Serial No.): _____	(Filing Date): _____	(Status): _____
(Application Serial No.): _____	(Filing Date): _____	(Status): _____
(Application Serial No.): _____	(Filing Date): _____	(Status): _____
(Application Serial No.): _____	(Filing Date): _____	(Status): _____

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.
 (List name and registration number)

Name: Donald R. Boys

Reg. No. 35,074

SEND CORRESPONDENCE TO:

Donald R. Boys
 P O. Box 187
 Aromas, CA 95004

DIRECT TELEPHONE CALLS TO:

Donald R. Boys (831) 726-1457

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full name of sole or first inventor: Mario Nemirovsky

1st inventor's signature: [Signature] Dated: 06/23/2000
Residence: 19750 Northampton Drive, Saratoga, CA, 95070 Citizenship: USA
Post Office Address: Same

Full name of 2nd joint inventor, if any: Narendra Sankar

2nd inventor's signature: [Signature] Dated: 06/23/2000
Residence: 450 Oak Grove Drive, Apt. #313, Santa Clara, CA, 95054 Citizenship: India
Post Office Address: Same

Full name of 3rd joint inventor, if any: Adolfo Nemirovsky

3rd inventor's signature: [Signature] Dated: 06/23/2000
Residence: 1044 Alderbrook Lane, San Jose, CA, 95129 Citizenship: USA
Post Office Address: Same

Full name of 4th joint inventor, if any: Enne Musoll

4th inventor's signature: [Signature] Dated: 06/23/2000
Residence: 7210 VIA ROMERA, SAN JOSE CA 95139 Citizenship: Spain
Post Office Address: Same

Full name of 5th joint inventor, if any: _____

5th inventor's signature: _____ Dated: _____
Residence: _____ Citizenship: _____
Post Office Address: _____

Full name of 6th joint inventor, if any: _____

6th inventor's signature: _____ Dated: _____
Residence: _____ Citizenship: _____
Post Office Address: _____

Full name of 7th joint inventor, if any: _____

7th inventor's signature: _____ Dated: _____
Residence: _____ Citizenship: _____
Post Office Address: _____

Full name of 8th joint inventor, if any: _____

8th inventor's signature: _____ Dated: _____
Residence: _____ Citizenship: _____
Post Office Address: _____